

NASA-CR-191695

R & D Status Report No. 2  
Low Voltage Arc Head Vacuum Welding  
Report Period: October 9 through December 21, 1992

Contract No. NAS8-39358

Prepared for  
NASA Marshall Space Flight Center  
AL, 38812

(NASA-CR-191695) LOW VOLTAGE ARC  
HEAD VACUUM WELDING Interim Status  
Report No. 2, 9 Oct. - 21 Dec. 1992  
(Electric Propulsion Lab.) 33 p

N93-71234

Unclassified

Z9/33 0139637

Electric Propulsion Laboratory, Inc.  
Monument, Colorado  
December 22, 1992

## Project Summary

This project supports NASA, Marshall Space Flight Center, by developing a low voltage arc head for space welding applications. Attachment I to this report documents the Statement of Work (SOW) for this effort. The following sections describe the progress made by EPL during this reporting period towards the completion of the tasks in this SOW.

### Task 1

Work on upgrading the archead power system was completed. The power system design was modified from that reported in the previous status report so that arc current control is possible in current increments of 1.0 A.

All components for the precision X,Y,Z arc head melt tracking test fixture were procured and delivered to EPL. Fabrication of this test fixture was initiated. To ensure accurate motion of the arc head, this test fixture is being fabricated while affixed to a large EPL granite surface table in order that alignment between the three arc head position axes is preserved. Attachment II shows the design and tolerance specifications of the open pillow block bearings, and supporting T bar and bearing guide rails, which EPL is using to fabricate the X,Y,Z arc head tracking test fixture. This positioning system was selected by EPL because of its accuracy, rigidity and excellant tolerance to continuous operation in a vacuum environment.

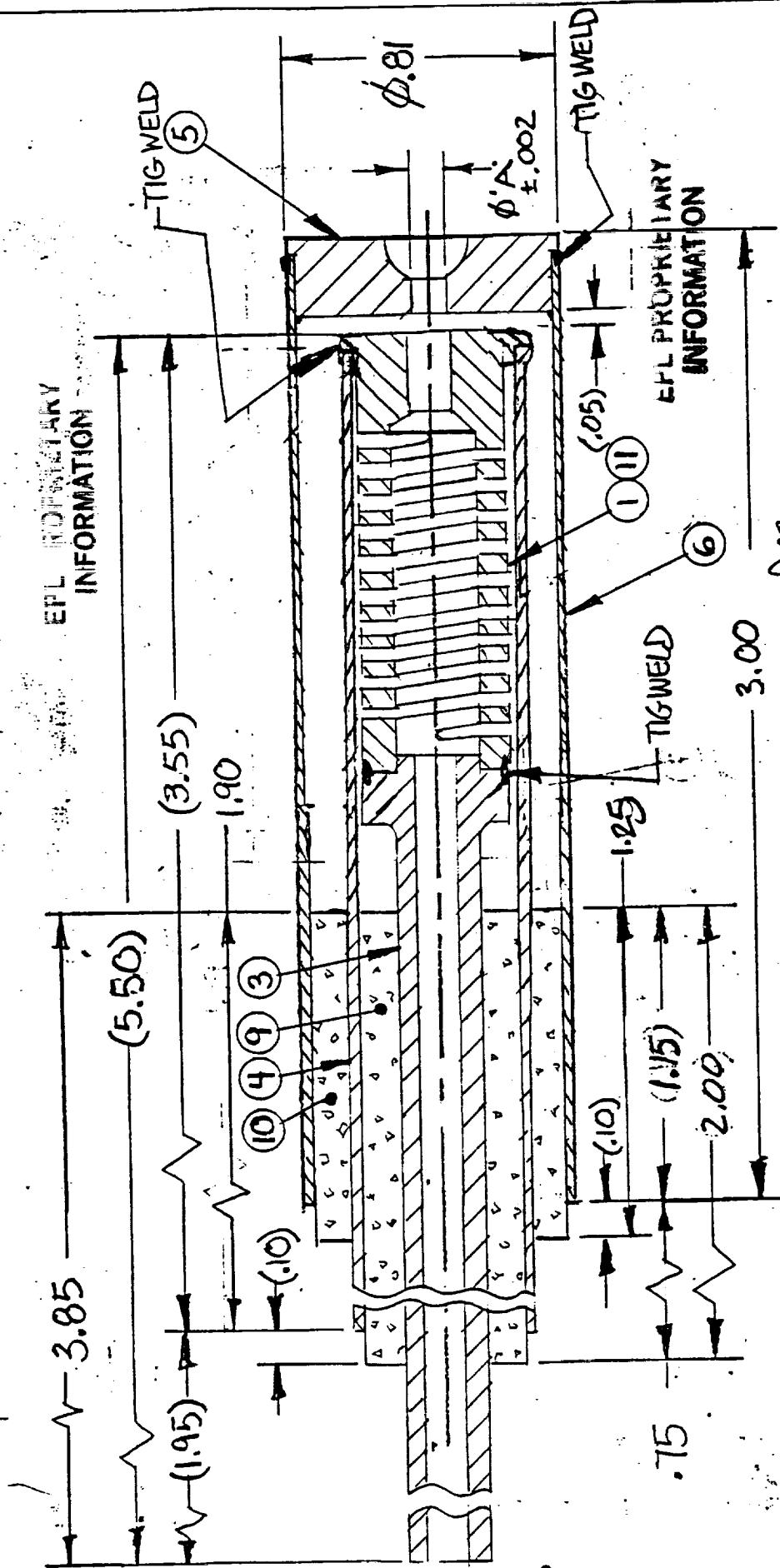
Work on developing the data acquisition software to support the needs of the arc head welding development program was largely completed. Attachment III lists the present compiled version of this code. This software was written in Microsoft C.

### Task 2

Fabrication of two prototype low voltage arc heads was completed by the vendor, Spectra-Mat., Inc., and these arc heads were delivered to EPL. Figures 1a - 1h show the machining drawings prepared by this fabrication vendor from the original EPL drawings presented in the previous status report.

Figures 2a - 2d show photographs of the two prototype arc heads. Figure 2a shows these arc heads side by side with a nickel for size comparison. It should be noted that this arc head design will be tested with, and without, an applied magnetic field. Without an applied magnetic field, the arc head largest outside diameter will be as shown in Fig. 2a. If the complexity of adding a permanent magnet is justified on the basis of significant improvement in arc head weld efficiency, then the addition of such a magnet would increase the arc head outside diameter by about a factor of two to three from that shown in Fig. 2a.

DASH NO	'A' DIA	REVISIONS	PART NO	APPROVED
REV	DESCRIPTION	EDD	DATE	
A	.062	102-630,102-631,102-632,102-633 102-637 TO REVA, J08 BECAME A PROTOTYPE	IN PROCESS 920105	10/30/92 BL
O1	.100			



UNLESS OTHERWISE SPECIFIED	DATE
DIMENSIONS ARE IN INCHES	
XXX ± .005 - .008 ± .02	DRAWN BY BL 10/21/92
FINISH 63 MICROINCH	CHK KM 6 10/22/92
ANGLES 9 DEG. 30 MIN.	ENGRS
EDGES AND CORNERS	
.005 R MAX	
MATERIAL	CHK

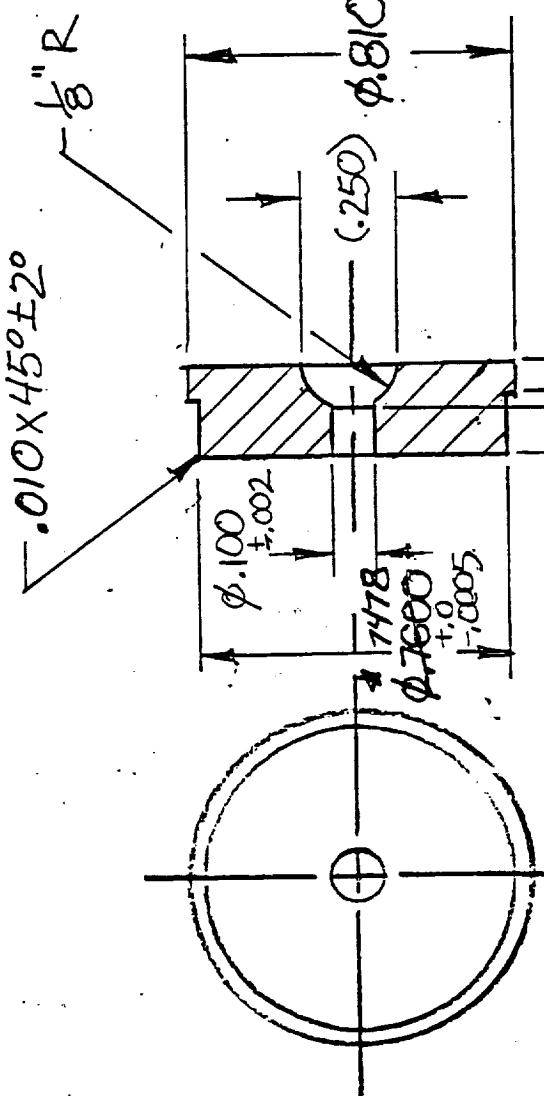
**ORIGINAL PRINT**  
(WHEN IN RED)

SIZE	FSCH NO.	REV
A	102-627	A
SCALE 2:1	920105	SHEET 1 OF 2

Fig. 1a Arched assembly drawing.

REV	DESCRIPTION	REVISIONS		PART NO 102-632	
		EDD	DATE	APPROVED	
A	$\phi .7600$ WAS.7495, $\phi .810$ WAS.750	IN PROCESS 920105	10/30/92	BK	

EPL PROPRIETARY  
INFORMATION



EPL PROPRIETARY  
INFORMATION

.055  
.250

\* Change  
per BK/TT  
10/3/92

SPECTRA-MAT. INC.  
NOZZLE INSERT

10/19/92

DATE

UNLESS OTHERWISE SPECIFIED  
DIMENSIONS ARE IN INCHES

XXX ± .005	YX ± .02	DRAWN	BK	10/19/92
FINISH 63 MICRON	ANGLE 0 DEG.	CK	KMG	10/19/92
ANGLES 0 DEG.	30 MIN	ENRG		
EDGES AND CORNERS				
.005 R MAX				
MATERIAL				
84% DENSE				
TUNGSTEN				

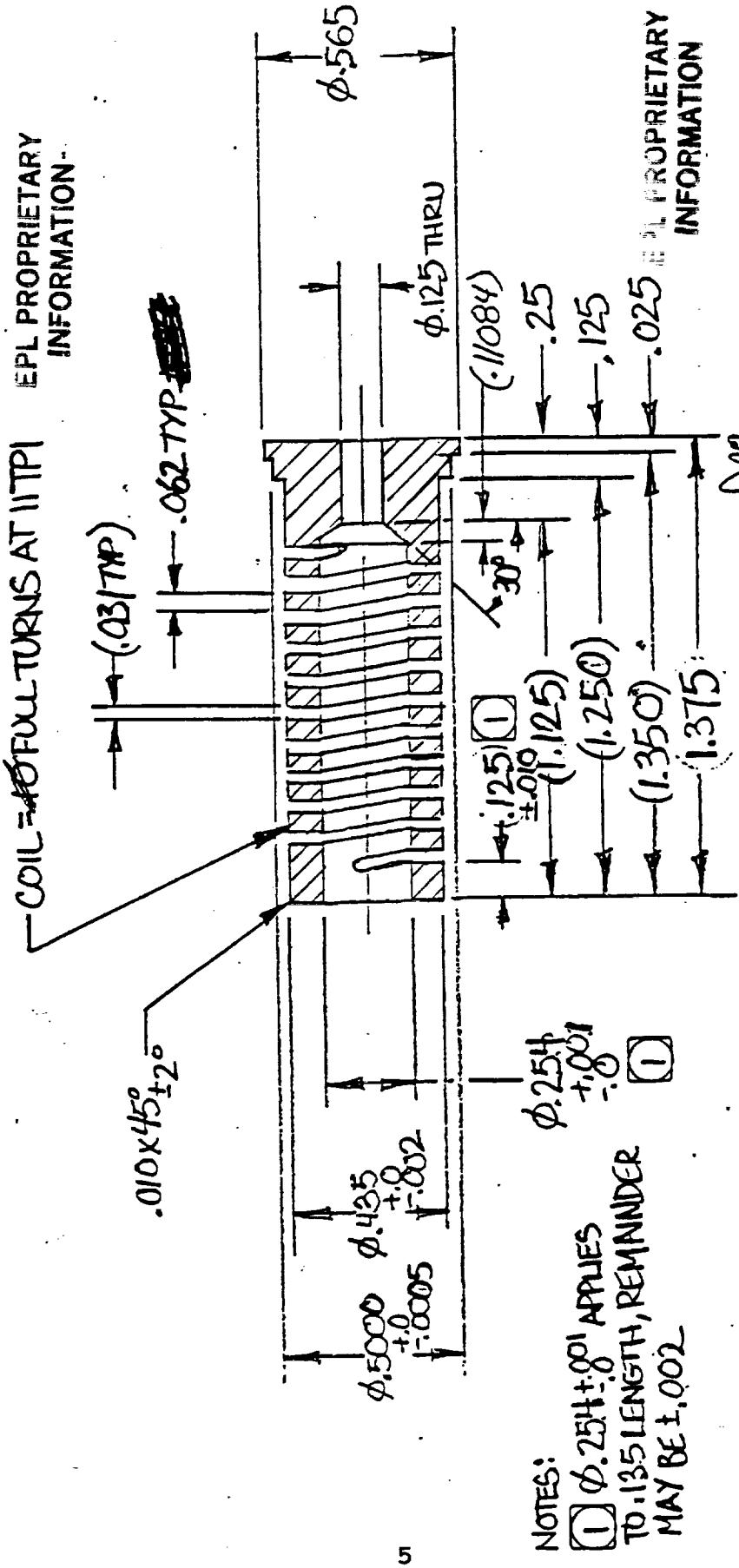
REV	ORIGINAL PRINT (WHEN IN RED)	SIZE	FECH NO.	DOC. NO.
A	102-632	4:1	920105	SHEET 1 OF 1

Fig. 1b Detail of arc forming orifice plate.

REVISIONS		PART NO 102-628		
REV	DESCRIPTION	ECO	DATE	APPROVED

8/11/1892  
9

EPL PROPRIETARY  
INFORMATION

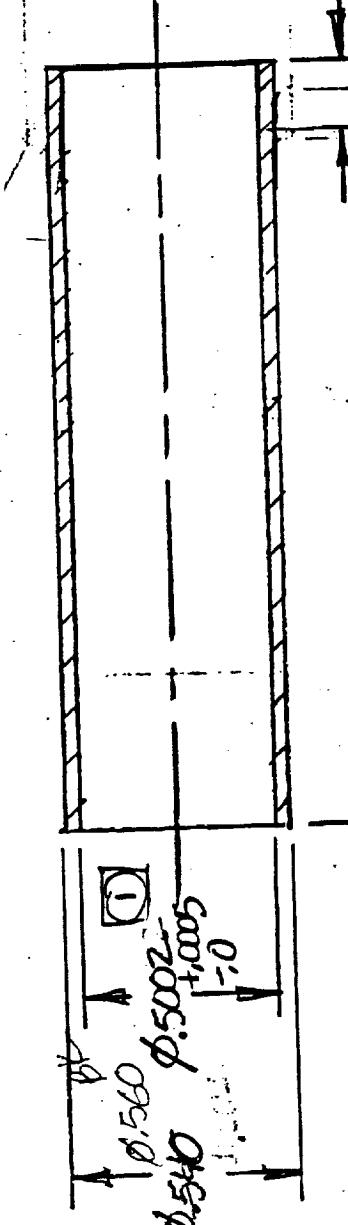


REF ID ES	DRAWN BY CLIK	DATE 10/19/92	SPECRA-MAT. INC.	REV -
MIN	ENGRG	10/22/92	EMITTER COIL	
CLIK				
ORIGINAL PRINT (WHEN IN RED)		SIZE A FSCH NO.	ORG. NO. 102-628	SHEET 1 OF 1
		SCALE 7:1	SCALE 970/05	

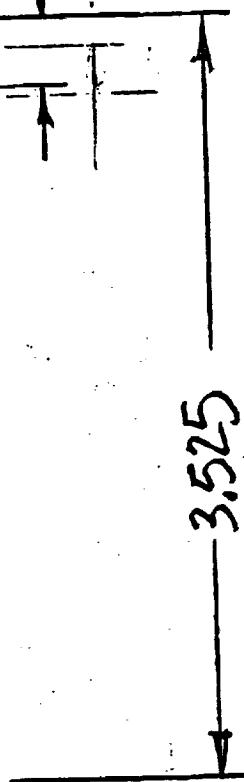
Fig. 1c Helical electron emitter insert.

REV	DESCRIPTION	REVISIONS	PART NO	APPROVED
A	Ø.540 WAS.540±.002 NOTE: +.001 WAS ±.002	In Process	19/80/102	BK

R. 212 PROPRIETARY INFORMATION



.040 MIN (i)



3.525

PROPRIETARY INFORMATION

① Ø.5002 ±.0005  
APPLIES TO Ø.5002 LENGTH,  
BALANCE IS ±.001  
(Fit to insulation)

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES				DATE	SPECTRA-MAT. INC.
XXX ± .005	.XX ± .02	DRAWN	BK	10/09/92	
FINISH 63	MICROINCH	CHK	L446	10/22/92	
ANGLES 0 DEG.	EDGES AND CORNERS	ENGRG			STEM
.005 R MAX		CHK			
MATERIAL					

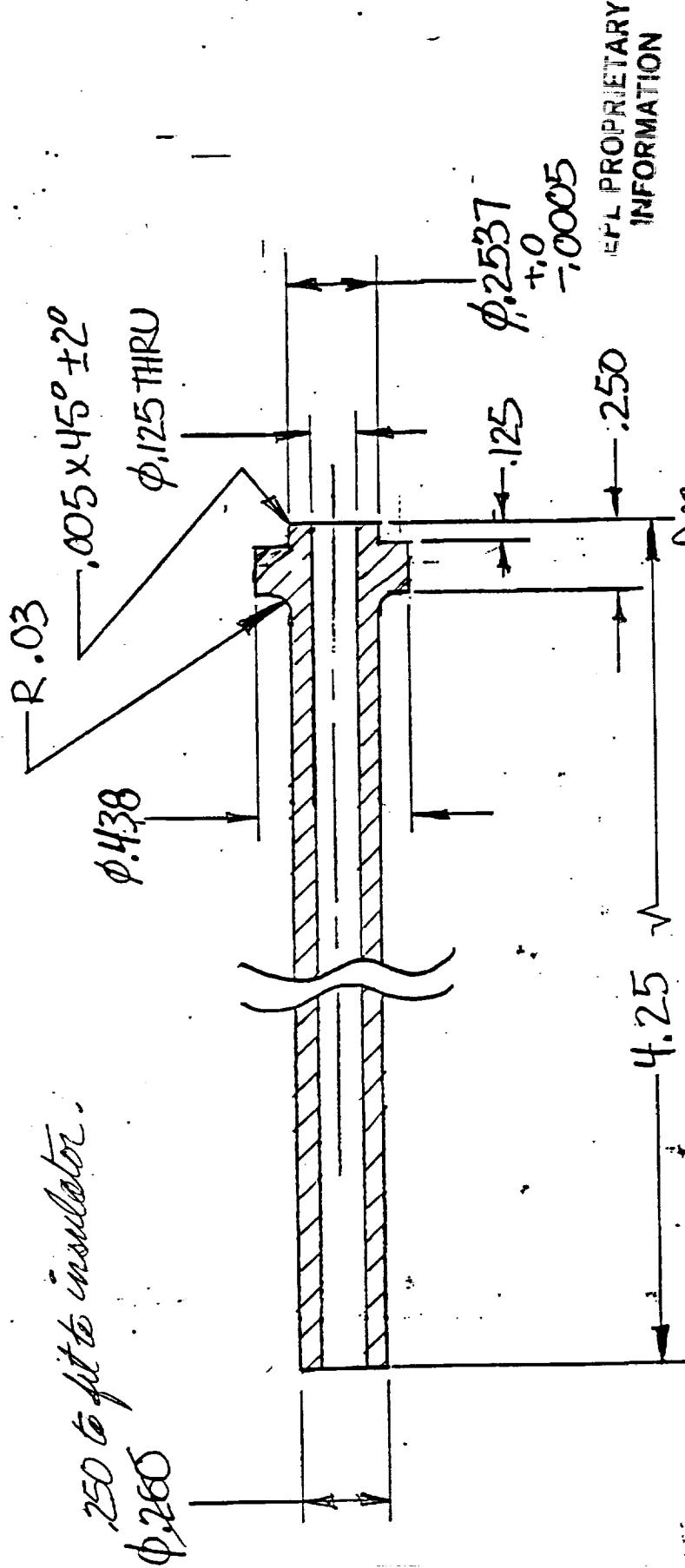
ORIGINAL PRINT (WHEN IN PDF)	SIZE	FSM NO.	DATE	DIAG. NO.	REV
May	A	102-630	10/11/92	102-105	A

Fig. 1d outer body cover for helical insert.

REV	DESCRIPTION	EDD	DATE	APPROVED

REVISIONS

PART NO 102-624

EPL PROPRIETARY  
INFORMATION

UNLESS OTHERWISE SPECIFIED  
DIMENSIONS ARE IN INCHES

.005 ± .005	.XX ± .02	DRAWN	3K	DATE
FINISH 63 MICROINCH		CIRK	10/19/92	
ANGLES 0 DEG. 30 MIN		ENRG	10/20/92	
EDGES AND CORNERS				
.005 R MAX				

MATERIAL

SPECTRA-MAT. INC.

STEM, SUPPORT

ORIGINAL PRINT (WHEN IN RED)	SIZE A FSCH NO.	DATE NO. 102-629	REV
SCAL 2:1	920105	SHEET 10F1	

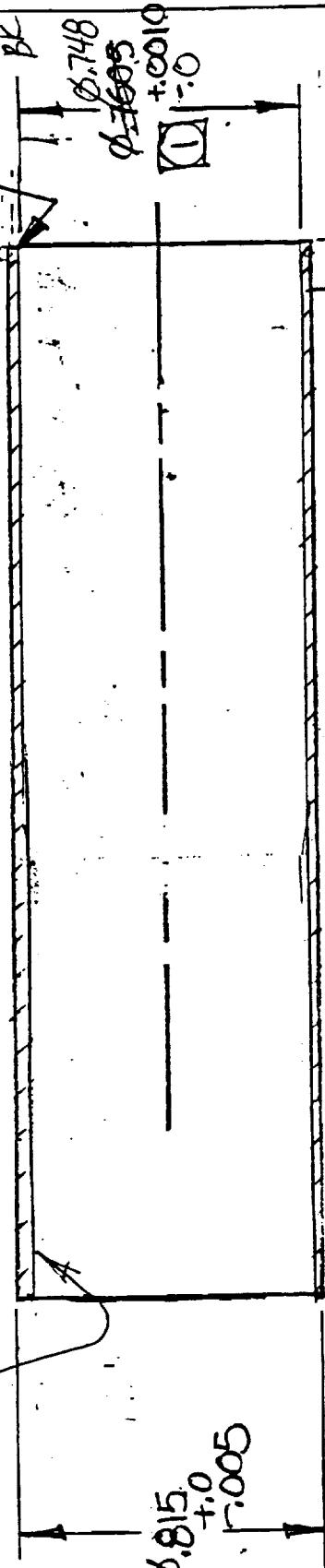
Fig. 1e Propellant injection tube.

REV	DESCRIPTION	REVISIONS		PART NO 102-633	
		ECO	DATE	APPROVED	
A	$\phi .815$ WAS .790, $\phi .7605$ WAS .750, NOTE 1 REWORKED, DEL 1.10 LENGTH AND TAPER.	IN PROCESS 920105	10/30/92	BK	

Fit of insulation on top

EPL PROPRIETARY  
INFORMATION

.00H-.006X45°±2°



Notes: .748 BK  
  $\phi .7605 \pm .005$  APPLIES  
 FOR .210 LENGTH, REMAINDER  
 MAY BE  $\pm .005$ .

REFERENCE  
INFORMATION

UNLESS OTHERWISE SPECIFIED  
DIMENSIONS ARE IN INCHES

XX ± .005	XX ± .02	DRAWN	BK	DATE
FINISH 63 MICROINCH		CHK	KM15	10/19/92
ANGLES 0 DEG.	30 MIN			10/22/92
EDGES AND CORNERS				
.005 R MAX				

MOLY

ORIGINAL PRINT  
(WHEN IN RED)

SIZE FSCN NO. DMC. NO. 102-633  
A

SCALE 2:1 920105 SHEET 1 OF 1

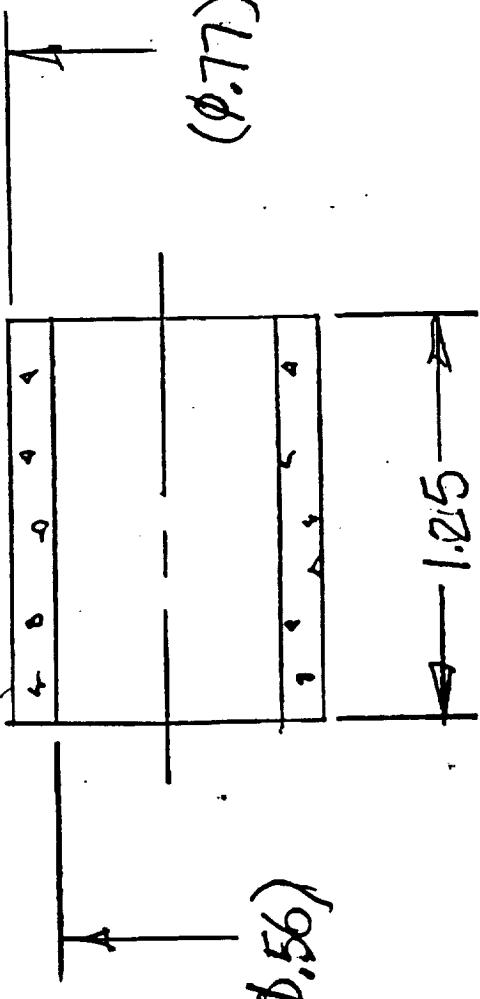
REV  
A

Fig. 1f Support tube for arc forming orifice plate.

REVISIONS		PART NO 102-637	
REV	DESCRIPTION	EDD	DATE APPROVED
A	(Ø.56) WAS( .54 ) (Ø.77) WAS( .75 )	IN PROCESS 920105	10/30/92 BK

Dia. ground to fit.

CONFIDENTIAL INFORMATION



CONFIDENTIAL INFORMATION

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES		DATE	SPECTRA-MAT. INC.
.000 ± .003	.00 ± .02	DRAWN BY KMG	10/20/92
FINISH 63 MICROINCH		CHK	10/22/92
ANGLES 0 DEG. 30 MIN		ENGRG	
EDGES AND CORNERS			
.005 R MAX			
MATERIAL			
Al <sub>2</sub> O <sub>3</sub>			
CUST. SUPPLIED			

(WHEN IN RED)

SCALE 211 920105

SHEET 1 OF 1

REV A

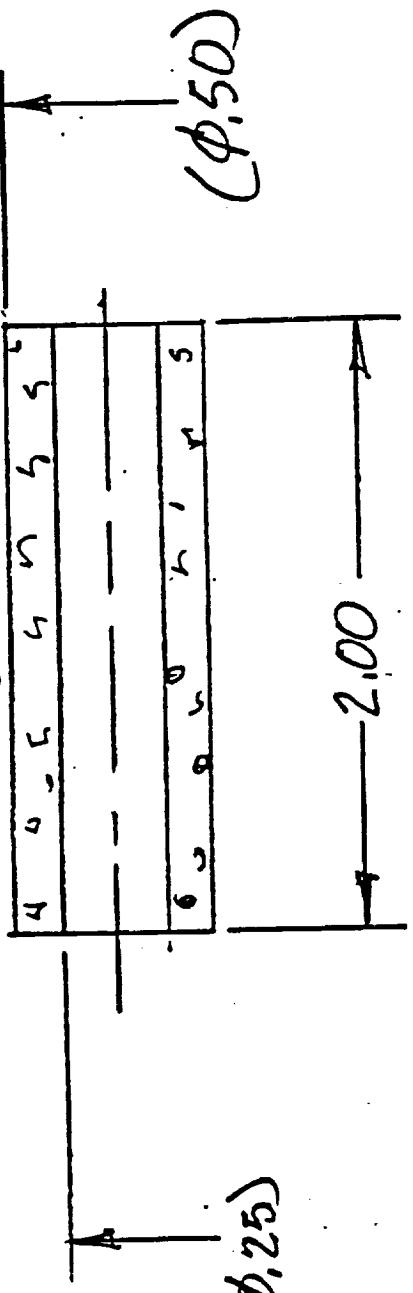
1q

Helical insert-to-outer body cover insulator tube

REV	REVISIONS	PART NO		102-636
	DESCRIPTION	ECO	DATE	APPROVED

Weld fit  
Downd Bk  
ground

EPL PROPRIETARY  
INFORMATION



PROPRIETARY  
INFORMATION

Fig. 1h Outer body cover-to-arc forming orifice plate support tube insulator.

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES		DATE	SPECTRA-MAT, INC.
.000 ± .005	.000 ± .002	DRAWN BY	10/22/92
FINISH 63 MICROINCH		CHK	KMG
ANGLES @ 0 DEG. 30 MIN		ENGRS	10/22/92
EDGES AND CORNERS			INSULATOR
.005 R MAX			
MATERIAL	CHK		
Al <sub>2</sub> O <sub>3</sub>		ORIGINAL PRINT (WHEN IN RED)	REV
CUST. SUPPLIED			

EPL PROPRIETARY  
INFORMATION

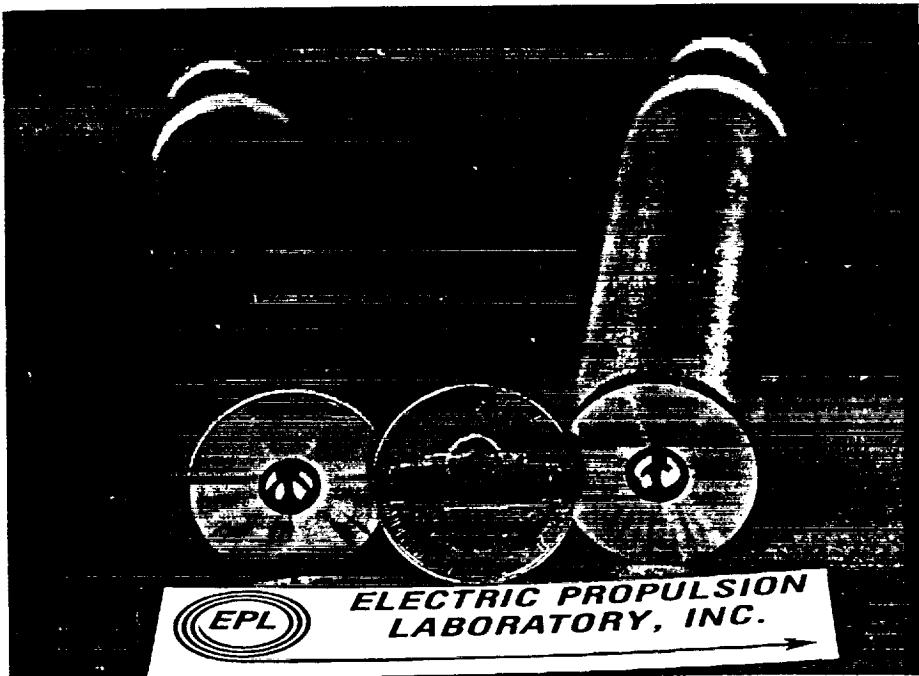


Fig. 2a Prototype low voltage arc heads. Note different arc forming orifices and nickel for size comparison.

EPL PROPRIETARY  
INFORMATION

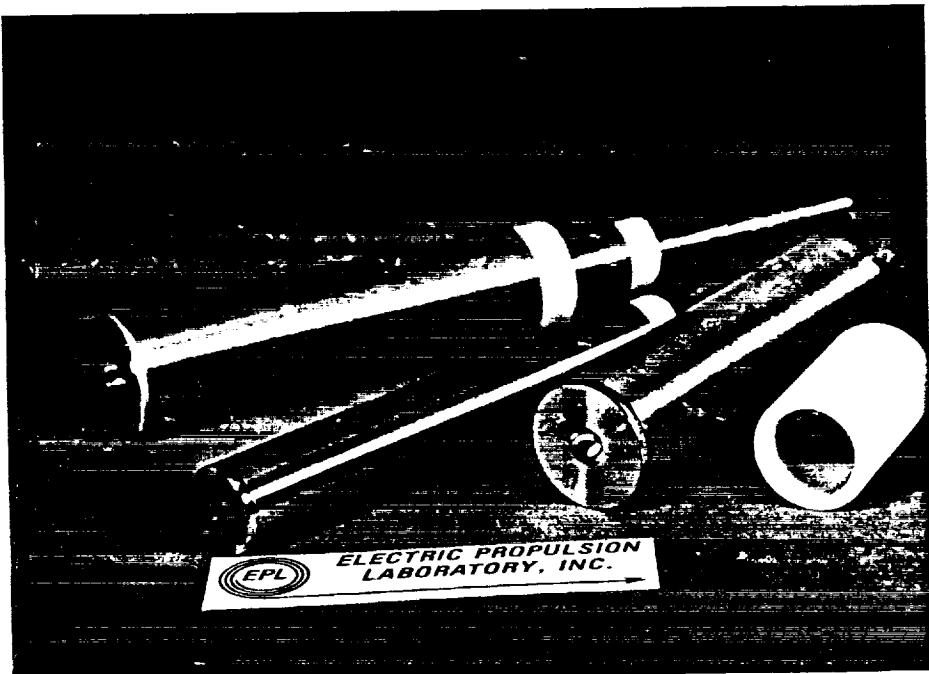


Fig. 2b Coaxial design of prototype arc heads and modular design permitting disassembly and inspection of critical components.

ORIGINAL PAGE  
BLACK AND WHITE PHOTOGRAPH

EPL PROPRIETARY  
INFORMATION



Fig. 2c Close-up of small diameter orifice arc head showing details of critical component surfaces prior to testing.

EPL PROPRIETARY  
INFORMATION

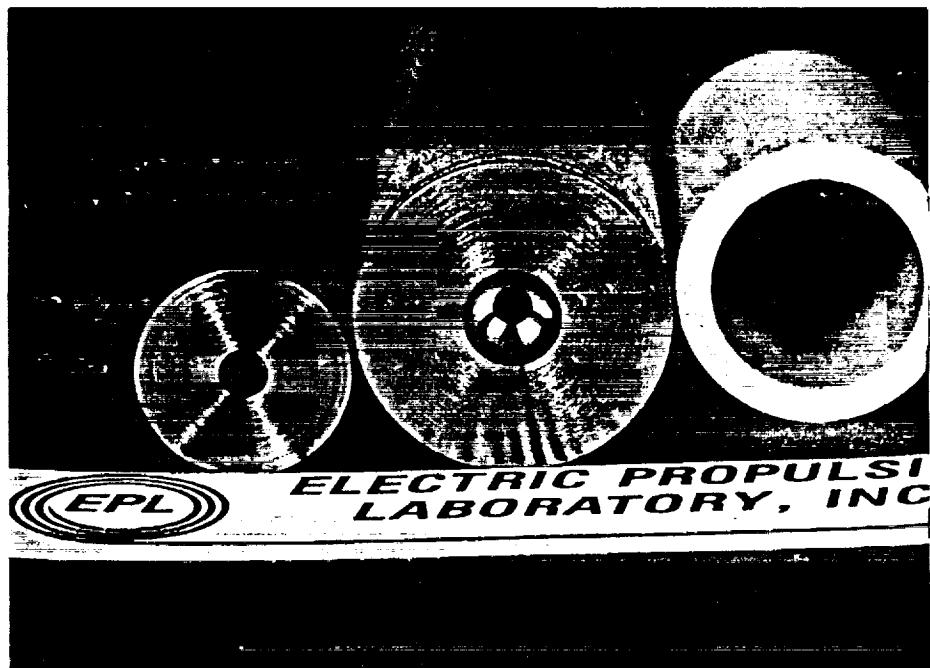


Fig. 2d Close-up of large diameter orifice arc head showing details of critical component surfaces prior to testing.

Figure 2b shows how these prototype arc heads can be partially disassembled for wear inspection of critical components following arc melt tracking evaluation tests. As can be seen from this photograph, the arc head design utilizes a triaxial utility feed system. The central molybdenum tube carries the arc head argon propellant and also serves as the helical electron emitting insert heater current return path. The intermediate molybdenum tube carries the input heater current to the helical electron emitting insert and also serves as the arc head electrical common point. The outer molybdenum tube supports the arc initiation and keep alive electrode which also serves as the arc shaping and primary radiant heat rejection, and arc head cooling, component.

Figures 2c and 2d show close-up details of the likely wear sites of the small and large diameter arc forming orifice designs respectively. Comparison with similar close-up photographs following arc head testing will allow for a better understanding of significant arc head erosion characteristics. This information will be valuable in guiding further design refinements for inclusion in later arc heads to be fabricated during this program.

The above information completes the requirements of this task.

### Task 3

No work was performed on this task.

### Task 4

No work was performed on this task.

### Task 5

No work was performed on this task.

### Task 6

No work was performed on this task.

### Task 7

No work was performed on this task.

### Current Problems

No problems exist to impede the progress of this contract.

### Future Work

Testing of the two prototype arc heads described in this report will be performed.

**CONTRACT NAS8-39358 EXPENDITURE SUMMARY**

Cumulative costs as of 12/16/92	\$41,652.86
Estimate of costs to complete	\$166,020.14
Estimate of percentage of physical completion	20%

## Attachment I

### Statement of Work:

EPL will perform the following tasks in Phase II to develop the Phase I proof-of-concept vacuum welding arc head into a high performance welding system adaptable to the space welding of tubular structures:

1. A vacuum test fixture will be assembled to allow for continuous arc melt tracking experiments to be performed for test durations of several minutes. This test fixture will be designed to permit arc length variations during arc head operation in addition to variations in melt track and/or weld velocity. In addition, EPL will upgrade the arc head power system and vacuum test facility to accommodate continuous arc currents up to 120 A. EPL will verify proper operation of the test fixture and power system using the Phase I proof-of-concept arc head.
2. EPL will design and fabricate a test bed arc head which will include provision for low voltage start-up, ie. less than 80 V. This arc head will be modular in design to permit parametric arc head geometry changes to be investigated. In addition, this test bed arc head will be configured to allow for application of a variable magnetic field for investigation of increased arc melt zone energy deposition.
3. Using the test fixture in Task 1, the modular test bed arc head fabricated in Task 2 will be operated over an arc power range of 1 - 2 kW for continuous periods of several minutes in duration and data will be collected to characterize the arc start-up, voltage, current, stability, shutdown, efficiency of energy transfer into the melt zone, melt zone focusing behavior, arc head thermal control requirements, propellant gas options, effect of weld sample contamination, and the arc head degradation and/or life limiting processes that occur during repeated testing. During these tests EPL will investigate the effects of power system ripple and arc current regulation on the melt zone quality. Correlations will be performed by EPL relating these measured test outputs and arc head geometries to maximize arc head performance in the power range 1 - 2 kW.
4. EPL will fabricate an arc head in accordance with the optimum geometrical design guidelines from the test results and analysis results from Task 3. EPL will operate this arc head to validate these performance predictions and operating characteristics.
5. From the results of Tasks 3 and 4, EPL will design and fabricate a plasma arc head configured to minimize clearance for tube welding applications.

6. EPL will assemble a simple tube welding test fixture for evaluation of the reduced clearance arc head fabricated in 5. Using this fixture, EPL will perform sample tube welds and evaluate these welds as a function of various plasma arc head operating conditions.
7. Based upon the results from Task 6, EPL will fabricate one complete arc head assembly for delivery to the Marshall Space Flight Center (MSFC). After fabrication, but prior to delivery, EPL will briefly operate this arc head to verify performance and to provide specific operating documentation.

# simplicity

PILLOW BLOCKS

pacific  
bearing

# Simplicity self-aligning pillow blocks bring three times the precision to bearing installation.

With self-alignment in all directions, Simplicity pillow block housings easily accommodate minor bore misalignment—a real benefit in terms of minimized machining/assembly time and installation cost. But unlike industry standard pillow blocks, the Simplicity system doesn't sacrifice precision for self-alignment.

Simplicity pillow blocks hold critical centerline accuracies to within  $\pm 0.0010$ ". That's **one-third** that of industry standard which allows a design tolerance for total centerline height range of  $0.006$ ". That gives the user a built-in misalignment of up to  $0.006$ " per pillow block—the low friction of ball bearing bushings is essential just to compensate for the loose tolerances in the housing.

With Simplicity, the pillow block—not the bearing—provides self-alignment capability. (The bearing provides rigid support over

Fig. 1 1/2° equals .0087 inches per inch radius\* at the midpoint of the pillow block bore allows housed bearings to pivot 1/2° off center in all directions (as shown in Fig. 1). The precision fit between the midpoint ID of the pillow block and the OD of Simplicity bearings provides the rigidity necessary to support overhung loads, while the self-alignment capability automatically accepts misaligned shafts.

Linear bearing systems that don't have built-in self-alignment are subject to rough operation, advanced wear rates and greatly shortened service life.

Installation inaccuracies such as those shown in Fig. 3 are all resolved by

the operational advantages built into the Simplicity system.

Simplicity pillow blocks are extruded from Type 6061-T6 aluminum. All critical surfaces are precision machined. The pillow blocks are hard-anodized to provide exceptional rigidity and resistance to corrosion. The resulting linear bearing/mounting system lets manufacturers obtain the precise alignment they require at prices that are very competitive for industrial applications. All critical mounting and external dimensions interchange with industry standard pillow blocks.

Pacific Bearing can also provide pillow blocks with straight bore fixed diameters upon request. Consult factory.

Fig. 2. Centerline tolerances

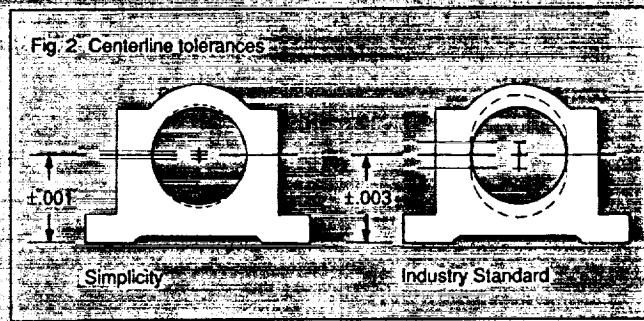
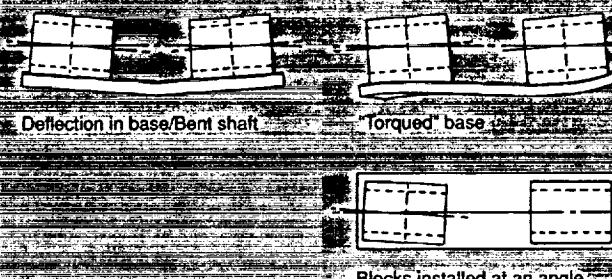


Fig. 3. Simplicity self-alignment solves these misalignment problems.



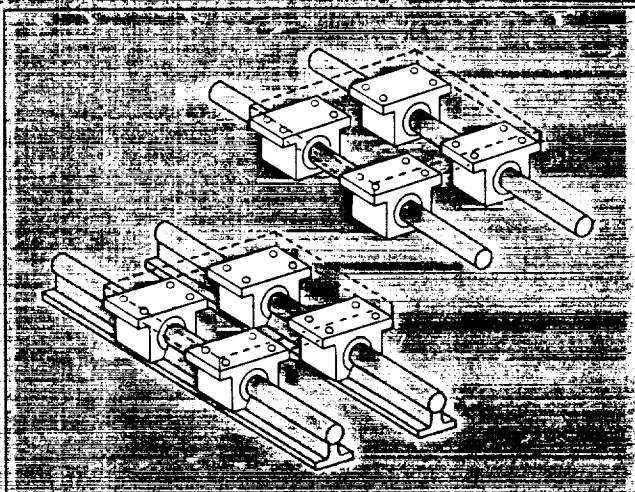
UNIFORMITY IS  
OF POOR QUALITY

Patent applied for



## Typical Applications

Simplicity parallel shaft installations as shown provide complete support and can be used with any tabletop arrangement that meets your shaft spacing requirements. Overhung loads are supported with three times the load-carrying capacity of linear ball bearings. Shaft scoring is eliminated.

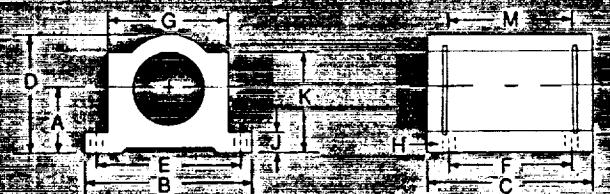


## Design Features

- Self-aligning to accept 1/2° misalignment off true centerline
- Heavy-walled hard-anodized Type 6061-T6 aluminum body
- Shaft centerline tolerances three times as precise as industry standard
- Optional sealed bearings for extremely dirty operations  
(See Table footnote)
- Completely interchangeable with industry standard units
- Simplified accurate alignment; easy mounting with no special mounting blocks
- Bearings held in place by internal snap rings

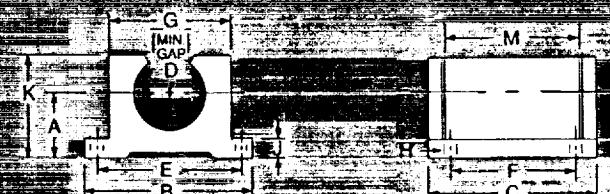
**Table 1**  
**Closed Series Pillow Blocks**

SIMPPLICITY PILLOW BLOCK NUMBER	BEARING BORE DIA. # .0010	A	B	C	D	E	F	G	H	J	K	M	MAX. LOAD RATING LBS. <sup>†</sup>	INTER- CHANGES WITH THOMSON PART NO.	
P-4	.250	.437	1-5/8"	1-3/16"	13/16"	.312	.750	1"	7/16"	5/32"	3/16"	3/4"	.750	300	SPB-4
P-6	.375	.500	1-3/4"	1-5/16"	15/16"	1.437	.875	1-1/8"	7/16"	5/32"	3/16"	7/8"	.875	510	SPB-6
P-8	.500	.687	2"	1-11/16"	1-1/4"	1.688	1.000	1-3/8"	7/16"	5/32"	1/4"	1-1/8"	1.250	975	SPB-8
P-10	.625	.875	2-1/2"	1-15/16"	1-5/8"	2.125	1.125	1-3/4"	7/8"	3/16"	9/32"	1-7/16"	1.500	1470	SPB-10
P-12	.750	.937	2-3/4"	2-1/16"	1-3/4"	2.375	1.250	1-7/8"	7/8"	3/16"	5/16"	1-9/16"	1.625	1905	SPB-12
P-16	1.000	1.187	3-1/4"	2-13/16"	2-3/16"	2.875	1.750	2-3/8"	10	7/32"	3/8"	1-15/16"	2.250	3525	SPB-16
P-20	1.250	1.500	4"	3-5/8"	3-13/16"	3.500	2.000	3"	10	7/32"	7/16"	2-1/2"	2.625	5145	SPB-20
P-24	1.500	1.750	4-3/4"	4"	3-1/4"	4.125	2.500	3-1/2"	1/4"	9/32"	1/2"	2-7/8"	3.000	7050	SPB-24
P-32	2.000	2.125	6"	5"	4-1/16"	5.250	3.250	4-1/2"	3/8"	13/32"	5/8"	3-5/8"	4.000	12525	SPB-32



**Table 2**  
**Open Series Pillow Blocks**

SIMPPLICITY PILLOW BLOCK NUMBER	BEARING BORE DIA. # .0010	A	B	C	D	E	F	G	H	J	K	M	MAX. LOAD RATING LBS. <sup>†</sup>	INTER- CHANGES WITH THOMSON PART NO.	
P-8-OPN	.500	.687	2"	1-11/16"	5/16"	1.688	1.000	1-3/8"	7/16"	5/32"	1/4"	1-1/8"	1.250	975	SPB-8-OPN
P-10-OPN	.625	.875	2-1/2"	1-15/16"	3/8"	2.125	1.125	1-3/4"	7/8"	3/16"	9/32"	1-7/16"	1.500	1470	SPB-10-OPN
P-12-OPN	.750	.937	2-3/4"	2-1/16"	7/16"	2.375	1.250	1-7/8"	7/8"	3/16"	5/16"	1-9/16"	1.625	1905	SPB-12-OPN
P-16-OPN	1.000	1.187	3-1/4"	2-13/16"	9/16"	2.875	1.750	2-3/8"	10	7/32"	3/8"	1-15/16"	2.250	3525	SPB-16-OPN
P-20-OPN	1.250	1.500	4"	3-5/8"	5/8"	3.500	2.000	3"	10	7/32"	7/16"	2-1/2"	2.625	5145	SPB-20-OPN
P-24-OPN	1.500	1.750	4-3/4"	4"	3/4"	4.125	2.500	3-1/2"	1/4"	9/32"	1/2"	2-7/8"	3.000	7050	SPB-24-OPN
P-32-OPN	2.000	2.125	6"	5"	1"	5.250	3.250	4-1/2"	3/8"	13/32"	5/8"	3-5/8"	4.000	12525	SPB-32-OPN



<sup>†</sup> MAXIMUM SPEED 400 FT/MIN, MAXIMUM PV 10,000 PSI/FT/MIN; SEE SIMPLICITY BEARING CATALOG PAGE 5 FOR DETAILS

<sup>†</sup> MAXIMUM LOAD MUST BE REDUCED BY 1/2 WHEN LOAD IS APPLIED TO THE OPEN HALF OF THE BEARING

SEALS: STANDARD PILLOW BLOCKS DO NOT HAVE SEALS. SEALS INCREASE FRICTION AND ARE NEEDED ONLY

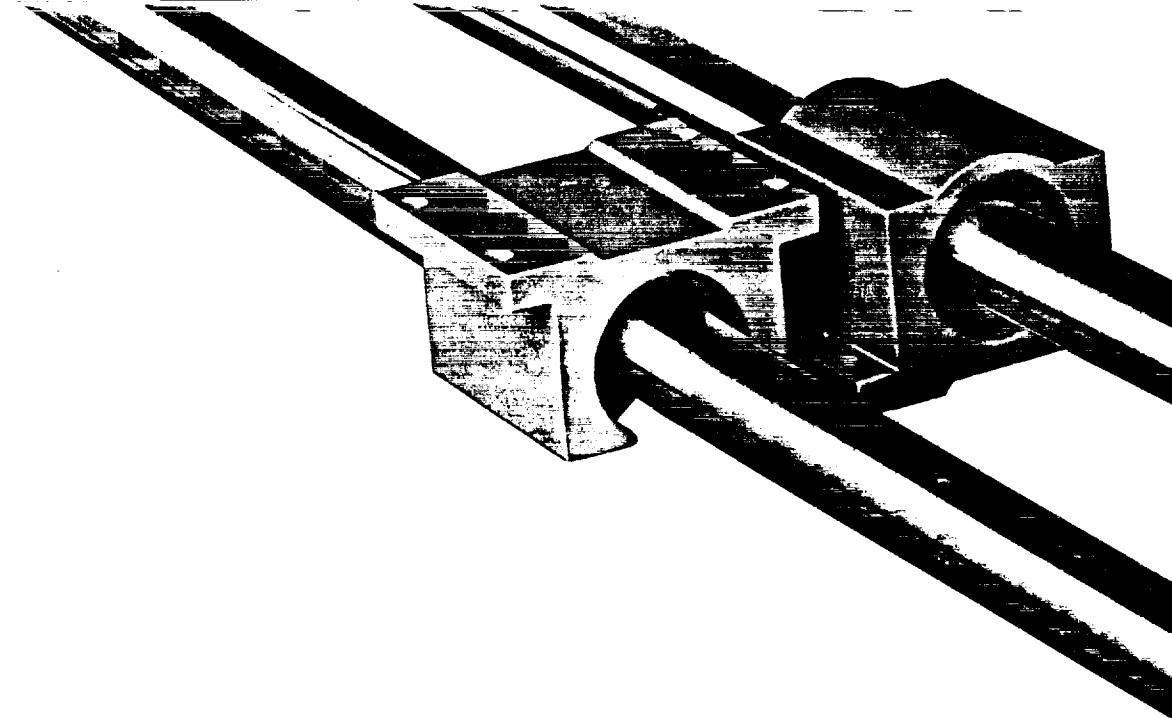
IN THE MOST CONTAMINATED ENVIRONMENTS. SEALS ARE INSTALLED INTO THE ENDS OF THE BEARINGS.

ORDERING SEALS: ADD THE LETTERS -DS (MEANING DOUBLE SEALS)

TO THE END OF THE PART NUMBER

EXAMPLE: A 1" SHAFT SIZE CLOSED PILLOW BLOCK WITH SEALS

WOULD BE P-16-DS



**STANDARD WARRANTY:** Pacific Bearing Co. guarantees that any product of its manufacture, when upon examination by a Pacific Bearing authorized representative is found to be defective in either workmanship or material, and whereby it is not suitable under proper usage and service for the purpose for which designed, will be replaced free of charge including transportation but not including the cost of installation. Pacific Bearing Co. assumes no other obligation or liability in connection with its use or the user's end product.

**pacific**  
                  
**bearing co.**

200 Quaker Road • Rockford, IL 61104  
P.O. Box 6980 • Rockford, IL 61125-6980  
815-962-8979 • FAX 815-962-3818

1 800 962 8979

### Attachment III

```

/* Program Weld.c
 * First: Set up screen display
 * Second: initialize OPTOMUX system: set up COM port, power up clear
 * Third: Enter module read loop which includes automated data and
 *         user inputted data
 * Fourth: At top of each data read loop, check for user input and respo
 */

#define CLEAR "\x1B[2J"           /* clear screen, home cursor */
#define BANNER_TOP 1             /* screen positions for banner */
#define BANNER_LEFT 1
#define BANNER_BOT 10
#define BANNER_RIGHT 75
#define WINDO_TOP 11             /* screen positions for data display */
#define WINDO_LEFT 1
#define WINDO_BOT 25
#define WINDO_RIGHT 75
#include <stdio.h>
#include <ctype.h>
#include <time.h>
#include <graph.h>
#include <conio.h>
#include <math.h>
#include <float.h>
int near errors;
int near address;
int near command;
int near modules[16];
int near modifers[2];
int near info[16];
int near infot[16];
void far pascal optoware(int near*, int near*, int near*,
                         int near*, int near*, int near*);

main()
{
    /* counters */
    int read_no, j, module_no;

    /* control flags */
    char stop, go, analyse, analyse1, analyse2, key, key2, print_now;

    /* input-output buffers */
    char tmpbuf[136], user_text[128], timbuf[128];

    /* time variables */
    time_t start_time, clock_time, elapsed_time;

    /* user control inputs, etc. */
    float read_interval, print_interval, time_min, user_data;

    /* data arrays */
    float convert[16];

    /* analysis variables, hard wired */
}

```

```

float grav, density, sgravty, h2oheat, egheat;

/* calorimetry variables, inputed by user via keyboard */
float cal_in, cal_out, cal_flow, pcal;

/* data input via modules */
float tank_press, arc_volt, arc_amp, keep_volt, keep_amp, heat_volt,
float x_position, y_position, z_position;

/* analysis variables, calculated */
float totpow, weld_effic, weld_speed, egfract, spheat, deltat;

/* windows structure */
struct rccoord oldpos;
struct videoconfig vc;

/* Initialize physical parameters as required */
density = 0.000;           /* TBD propellant density, gm/cc */
sgravty = 1.021;           /* coolant specific gravity */
h2oheat = 4203.2;          /* specific of water @ 38 deg C, j/kgC */
egheat = 2520.3;           /* specific heat of ethylene glycol @ 38 de

/* Initialize input parameters */
read_interval = 5.0;        /* read OPTOMUX every 5 minutes */
print_interval = 5.0;       /* print outputs every 5 minutes */
cal_in = 0.0;               /* plume calorimeter inlet temperature */
cal_out = 0.0;               /* plume calorimeter outlet temperature */
cal_flow = 0.0;              /* plume calorimeter coolant flow, gpm */

/* Initialize screen */
_clearscreen(_GCLEARSCREEN);
_setvideomode(_ERESCOLOR);
_getvideoconfig( &vc ); /* get video mode info */
_rectangle(_GBORDER, 0, 0, vc.numxpixels-1, vc.numypixels-1 );
_settextposition( 2, 34 );
_outtext("SPACE WELDING");
_settextposition( 3, 31 );
_outtext("Data Retrieval Program");

/* Initialize retrieval frequency parameter */
read_interval = 15.0;      /* read OPTOMUX every 15 minutes */

/* Display banner describing hot keys and initial frequencies */
_settextposition(4,3);
sprintf(tmpbuf, "===== MENU =====");
_outtext(tmpbuf);
_settextposition( 5, 5 );
_outtext("f1 = Start")           f2 = Set Read Freq   f3 = Calori
_settextposition( 6, 5 );
_outtext("f4 = Calorimeter Temp") f5 = Add Text     f10 = End")
_outtext("f7 = Exit");

/* Display banner for user inputs */
_settextposition(7,3);
sprintf(tmpbuf, "===== INPUTS =====");
_outtext(tmpbuf);

```

```

_settextposition( 8, 5 );
sprintf(tmpbuf, "Read mins = %5.2f", read_interval);
_outtext( tmpbuf );
_settextposition( 9, 5 );
sprintf(tmpbuf, "          Calorimeter: Inlet Temp = %5.2f   Outlet Temp
_outtext ( tmpbuf );

/* Display banner for data */
_settextposition(10, 3 );
sprintf(tmpbuf, "===== DATA =====");
_outtext(tmpbuf);

/* Set-up OPTOMUX */
/* Use two pass protocol */
errors = 0;
address = 0;
command = 100;
info[0] = 0;
optoware(&errors,&address,&command,&modules[0],&modifers[0],&info[0])
if (errors < 0)
{
    printf("\a");           /* alarm bell */
    printf("  OPTOMUX RETURN ERROR: protocol ");
}

/* Specify the serial port (COM3) */
errors = 0;
address = 0;
command = 102;
info[0] = 3;
optoware(&errors,&address,&command,&modules[0],&modifers[0],&info[0])
if (errors < 0)
{
    printf("\a");           /* alarm bell */
    printf("  OPTOMUX RETURN ERROR: set serial port ");
}

/* Configure the serial port */
errors = 0;
address = 0;
command = 104;
info[0] = 9600;
optoware(&errors,&address,&command,&modules[0],&modifers[0],&info[0])
if (errors < 0)
{
    printf("\a");           /* alarm bell */
    printf("  OPTOMUX RETURN ERROR: configure serial port ");
}

/* Do power up clear */
errors = 0;
address = 255;           /* address analog brain board 1 */
command = 0;              /* power up clear */
optoware(&errors,&address,&command,&modules[0],&modifers[0],&info[0])
if ( errors < 0 )
{
}

```

```

printf("\a");           /* alarm bell */
printf(" OPTOMUX RETURN ERROR: power up clear brain board 1 ");
}
errors = 0;
address = 254;          /* address analog brain board 2 */
command = 0;             /* power up clear */
optoware(&errors,&address,&command,&modules[0],&modifiers[0],&info[0])
if ( errors < 0 )
{
    printf("\a");           /* alarm bell */
    printf(" OPTOMUX RETURN ERROR: power up clear brain board 2 ");
}

/* Set up thermocouple module; this is on Brain Board 1 */
errors = 0;
address = 255;
command = 76;            /* sets up thermocouple modules for reading
modules[0] = 2;           /* position of thermocouple module */
info[0] = 4;              /* specifies type J thermocouple */
optoware(&errors,&address,&command,&modules[0],&modifiers[0],&info[0])
if (errors < 0)
{
    printf("\a");           /* alarm bell */
    printf(" OPTOMUX RETURN ERROR: set thermocouple type ");
}

/* Set up variables for data collection */
read_no = 0;              /* no reads yet */
go = 'n';                /* set to 'y' when user strikes f1 key */
stop = 'n';               /* set to 'y' when user strikes f10 key */
analyse1 = 'n';            /* set to 'y' when user enters cal. flow da */
analyse2 = 'n';            /* set to 'y' when user enters cal. temp da */
print_now = 'n';           /* set to 'y' to override print interval */

/* Read analog data -- read analog inputs */
while ( stop == 'n' )      /* loops until user hits f10 key */
{
    /* Process user requests */
    if ( kbhit() != 0 )    /* First check to see if there is user input */
    {
        key = getch();       /* Gets last keyboard character */
        if ( key == 0 )       /* Check for function key */
        {
            key2 = getch();   /* Specifies which function key */
            switch (key2)
            {

                case 59:         /* F1 key: begin data reads */
                if ( go == 'n' )
                {
                    time(&start_time);
                    go = 'y';
                }
                break;
            }
        }
    }
}

```

```

case 60:          /* F2 key: change read frequency */
    _settextposition(9,5);
    _outtext("");
    _settextposition(9,5);
    _outtext("Enter new read frequency (min): ");
    scanf("%f", &read_interval);
    _settextposition(9,5);
    sprintf(tmpbuf,"Read mins = %5.2f", read_interval);
    _outtext(tmpbuf);
    if (read_no != 0)
    {
        read_no = (float)elapsed_time/(read_interval*60.0);
    }
break;

case 61:          /* F3 key: Enter calorimeter coolant flow */
    _settextposition(9,72);
    _outtext("");
    scanf("%f", &user_data);
    cal_flow = user_data;           /* plume calorimeter fl */
    cal_flow = sgravty*(0.0633*cal_flow-0.0041667); /* gpm -> l/
    _settextposition(9,72);
    sprintf(tmpbuf, "%5.2f", user_data);
    _outtext(tmpbuf);
    if ( go == 'y' )
    {
        analyse1 = 'y';           /* calorimeter flow dat */
    }
break;

case 62:          /* F4 key: Enter calorimeter temperatures */
    _settextposition(9,38);
    _outtext("");
    scanf("%f", &user_data);
    cal_in = user_data;           /* plume calorimeter in */
    _settextposition(9,38);
    sprintf(tmpbuf, "%5.2f", user_data);
    _outtext(tmpbuf);
    _settextposition(9,58);
    _outtext("");
    scanf("%f", &user_data);
    cal_out = user_data;          /* plume calorimeter ou */
    _settextposition(9,58);
    sprintf(tmpbuf, "%5.2f", user_data);
    _outtext(tmpbuf);
    if ( go == 'y' )
    {
        analyse2 = 'y'; /* calorimeter temperature data available */
    }
break;

case 63:          /* F5 key: insert user text */
fflush(stdin);
_settextposition(22,1);

```

```

sprintf(tmpbuf,"COMMENT: \n");
_outtext(tmpbuf);
gets(user_text);
_settextposition(23,1);
sprintf(tmpbuf, "%s \n", user_text);
_outtext( tmpbuf );
sprintf(tmpbuf, "COMMENT: %s \r\n", user_text);
fputs(tmpbuf, stdprn);
fflush(stdin);
break;

case 68:           /* F10 key: terminate program */
_settextposition( 22,1 );
sprintf(tmpbuf, "
_outtext( tmpbuf );
_settextposition( 23,1 );
_outtext( tmpbuf );
_strerror ( timbuf );
sprintf(tmpbuf, "\nData acquistion terminated at %s ", timbu
_outtext( tmpbuf );
sprintf(tmpbuf, "Data acquistion terminated at %s \r\n\f", t
fputs(tmpbuf, stdprn);
stop = 'y';
break;

}                   /* terminates switch statement */
}                   /* terminates function key if statement */
}                   /* terminates keyboard entry if statement */

/* When data collection flag is set, read OPTOMUX */
if ( go == 'y' )    /* collecting data */
{
    /* Find current time and check to see if it is time to read dat
time( &clock_time );                                /* current time in seco
elapsed_time = clock_time - start_time; /* time since last read
if ( (elapsed_time>(read_interval*60.0*read_no)) || (read_no==0
{                                              /* time to read and print data */
    read_no += 1;      /* increment read_no */

    /* Zero out arrays for data collection from BB #2 */
    for (j=0; j<16; j++)
    {
        info[j]      = 0;
        convert[j]   = 0.0;
    }

    /* Initialize parameters for OPTOMUX, Brain Board #2 */
    errors      = 0;
    address     = 254; /* analog brain board 2 */
    command     = 37;  /* read analog inputs */
    modules[0]   = 0;  /* 1st, OPTOMUX AD7 (0-10v), ion gauge
    modules[1]   = 1;  /* 2nd, OPTOMUX AD7 (0-10v), arc volts
    modules[2]   = 2;  /* 3rd, OPTOMUX AD7 (0-10v), keeper volt
    modules[3]   = 3;  /* 4th, OPTOMUX AD7 (0-10v), heater volt

```

```

modules[4] = 4; /* 5th, OPTOMUX AD9T (0-50mv), arc amps
modules[5] = 5; /* 6th, OPTOMUX AD13T(0-100mv), keeper amps
modules[6] = 6; /* 7th, OPTOMUX AD13T(0-100mv), heater amps
modules[7] = 7; /* 8th, OPTOMUX AD6 (0-5 v), tacometer
modules[8] = 8; /* 9th, OPTOMUX AD6 (0-5 v), flow chan 1
modules[9] = 9; /* 10th,OPTOMUX AD6 (0-5 v), z axis
modules[10]=10; /* 11th,OPTOMUX AD6 (0-5 v), x axis
modules[11]=11; /* 12th,OPTOMUX AD6 (0-5 v), y axis
modules[12] =-1; /* reading 9 modules this time

/* Call OPTOMUX on Brain Board #2, and read data */
optoware(&errors,&address,&command,&modules[0],&modifiers[0],

/* Check for OPTOMUX errors, Brain Board #2 */
if (errors < 0 ) /* OPTOMUX error detected */
{
    sprintf(tmpbuf,"\\a  OPTOMUX RETURN ERROR: read data \\n");
    _outtext(tmpbuf);
}

/* Convert raw data in info array; first convert the digital
/* OPTOMUX output to a fraction of total scale */
for (j=0; j<9; j++)
{
    /* All data transmitted from the OPTOMUX is in the */
    /* range 0 - 4095; convert this integer data into a */
    /* fraction of the full scale */
    convert[j] = (((float)info[j])/((float)4095));

    /* Perform conversions specific to individual data */
    if ( j==0 ) /* Ion gauge */
    {
        convert[0] = convert[0]*10.0; /* for display only, no
                                       /* converts from fracti
        convert[0] = pow( 10.0, convert[0]-11.0 ); /* from
                                       /* Granville-Phillips i
                                       /* the 1mA scale; conve
        tank_press = convert[0]; /* for analysis */
    }

    if ( j==1 ) /* Arc volts */
    {
        convert[1] = convert[1]*10.0; /* converts from fracti
        convert[1] = convert[1]*10.0; /* voltage reading was
                                       /* factor of 10 -- TO B
        arc_volt   = convert[1]; /* used in analysis */
    }

    if ( j==2 ) /* Keeper volts */
    {
        convert[2] = convert[2]*10.0; /* converts from fracti
        convert[2] = convert[2]*10.0; /* voltage reading was
                                       /* factor of 10 -- TO B
        keep_volt  = convert[2]; /* used in analysis */
    }
}

```

```

if ( j==3 )          /* Heater volts */
{
    convert[3] = convert[3]*10.0; /* converts from fracti
    convert[3] = convert[3]*10.0; /* voltage reading was
                                   /* factor of 10 -- TO B
    heat_volt  = convert[3];      /* used in analysis */
}

if ( j==4 )          /* Arc amps */
{
    convert[4] = convert[4]*50.0; /* converts from fracti
    convert[4] = convert[4]*6.0;  /* must convert from mV
                                   /* this conversion has
                                   /* checked as of 10/30/
    arc_amp    = convert[4];      /* for analysis */
}

if ( j==5 )          /* Keeper amps */
{
    convert[5] = convert[5]*100.0;           /* converts f
                                              /* to mV */
    convert[5] = convert[5]*6.0;             /* converts f
                                              /* MUST BE VE
    keep_amp   = convert[5];                /* for analys
}

if ( j==6 )          /* Heater amps */
{
    convert[6] = convert[6]*100.0;           /* converts f
                                              /* to mV */
    convert[6] = convert[6]*6.0;             /* converts from mV
                                              /* MUST BE VE
    heat_amp   = convert[6];                /* for analys
}

if ( j==7 )          /* Tacometer */
{
    convert[7] = convert[7] * 5.0;           /* converts f
                                              /* to value f
    convert[7] = convert[7] * 20.0;          /* converts f
                                              /* cm/min */
    weld_speed = convert[7];                /* THIS MUST
}

if ( j==8 )          /* Propellant flow */
{
    convert[8] = convert[8] * 5.0;           /* converts f
                                              /* value from
    convert[8] = convert[8] * 20.0;          /* converts f
                                              /* 0 - 100 sl
    mdot       = convert[8];                /* for analys
}

if ( j==9 || j==10 || j==11 )    /* XYZ carriage */
{

```

```

        convert[9] = convert[9] * 5.0;          /* converts f
                                                /* value from
        convert[9] = convert[9] * 5.0;          /* conversion
                                                /* volts to c
        z_position = convert[9];              /* for analys

        convert[10]= convert[10] * 5.0;         /* see above
        convert[10]= convert[10] * 5.0;         /* REQUIRED *
        x_position = convert[10];              /* for analys

        convert[11]= convert[11] * 5.0;         /* see above
        convert[11]= convert[11] * 5.0;         /* REQUIRED *
        y_position = convert[11];              /* for analys
    }

}

/* Zero out arrays for data collection from BB #1 */
for (j=0; j<16; j++)
{
    info[j]      = 0;
    convert[j]   = 0.0;
}

/* Initialize parameters fof OPTOMUX, Brain Board #1 */
errors      = 0;
address     = 255; /* analog brain board 2 */
command     = 77; /* read temperature */
modules[0]  = 0; /* 1st, OPTOMUX Type J Thermocouple */
modules[1]  = 1; /* 2nd, OPTOMUX Type J Thermocouple */
modules[2]  = 2; /* 3rd, OPTOMUX Type J Thermocouple */
modules[3]  = 3; /* 4th, OPTOMUX Type J Thermocouple */
modules[4]  = -1; /* reading 4 temperatures */

/* Call OPTOMUX on Brain Board #1 and read data */
optoware(&errors,&address,&command,&modules[0],&modifiers[0],

/* Check for OPTOMUX errors, Brain Board #1 */
if ( errors < 0 )           /* OPTOMUX error detected */
{
    sprintf(tmpbuf,"\\a  OPTOMUX RETURN ERROR: read data \\n");
    _outtext( tmpbuf );
}

/* Convert raw temperature data; all temperature data are re
/* in 1/16th degree C so conversion is simply division by 16
for (j=0; j<5; j++)
{
    convert[j] = convert[j]/16.0;      /* convert from 1/16ths
                                         /* degrees C to degrees
}

/* Get current time */
    strftime( timbuf );             /* current time in hh:mm:ss */
    time( &clock_time );           /* current time in seco

```

```

elapsed_time = clock_time - start_time;           /* elapsed time
time_min = elapsed_time / 60.0;                  /* elapsed time

/* Evaluate input power */
totpow = arc_volt*arc_amp + keep_volt * keep_amp + heat_volt

/* Display data read automatically and parameters calculated
/* data read automatically */
_settextposition(12,3);
sprintf(tmpbuf,"          Time: %s             Elapsed Mins:
_outtext(tmpbuf);
_settextposition(13,3);
sprintf(tmpbuf,"Arc:      V(V)      = %4.1f    I(A)
_outtext(tmpbuf);
_settextposition(14,3);
sprintf(tmpbuf,"Keeper:   V(V)      = %4.1f    I(A)
_outtext(tmpbuf);
_settextposition(15,3);
sprintf(tmpbuf,"Heater:   V(V)      = %4.1f    I(V)
_outtext(tmpbuf);
_settextposition(16,3);
sprintf(tmpbuf,"Performance: Power(w) = %4.1f    Speed(c)
_outtext(tmpbuf);
_settextposition(17,3);
sprintf(tmpbuf,"          X(cm)     = %4.1f    Y(cm)
_outtext(tmpbuf);

/* If calorimetry data are available, calculate frozen flow
if ( analyse1 == 'y' && analyse2 == 'y' )
{
    /* Calculate power input to the calorimeter */
    egfract = 777.3*log(sgravty) + 1.571;           /*specific heat
    spheat = (1.0-(egfract/100.0))*h2oheat + (egfract/100.0)*
    deltat = cal_out - cal_in;                      /* change in
    pcal   = cal_flow * spheat * deltat;            /* power input
                                                       /* power in t

    /* Calculate weld efficiency as the ratio of the power */
    /* transferred to the calorimeter (i.e. target) to the */
    /* arched input power */
    weld_effic = pcal/totpow;

    /* Display calorimetry parameters */
    _settextposition(19,3);                           Elapsed Mins: %8.
    sprintf( tmpbuf,"Time: %s
    _outtext( tmpbuf );
    _settextposition(20,3);
    sprintf(tmpbuf,"Temperature: Inlet(C)           = %4.1f
    _outtext( tmpbuf );
    _settextposition(21,3);
    sprintf(tmpbuf,"Calorimetry: Target input power(w) = %4.1f
    _outtext(tmpbuf);

    /* Reset flags */
}

```

```

analyse1 = 'y';          /* assumes that flow is constant
analyse2 = 'n';          /* assumes that temperatures vary
print_now = 'y';         /* print calorimetry data now */
}

/* Print data on prescribed interval */
if ( elapsed_time > (print_interval*60.0*read_no) || (read_no ==
{
    fputs(tmpbuf, stdprn); /* output to printer */
    sprintf(tmpbuf, "\r\n"); /* skip a line on the printer */
    fputs(tmpbuf, stdprn);
    sprintf(tmpbuf, "      Time: %s           Elapsed Mi
    fputs(tmpbuf, stdprn);
    sprintf(tmpbuf, "    Arc:      V(V)      = %4.1f     I(
    fputs(tmpbuf, stdprn);
    sprintf(tmpbuf, "    Keeper:    v(V)      = %4.1f     I(
    fputs(tmpbuf, stdprn);
    sprintf(tmpbuf, "    Heater:    V(V)      = %4.1f     I(
    fputs(tmpbuf, stdprn);
    sprintf(tmpbuf, "    Performance: Power(w) = %4.1f     Sp
    fputs(tmpbuf, stdprn);
    sprintf(tmpbuf, "                  X(cm)    = %4.1f     Y(
    fputs(tmpbuf, stdprn);

    if (print_now = 'y' )
    {
        sprintf(tmpbuf, "    Temperature: Inlet(C)  = %6.3f
        fputs(tmpbuf, stdprn);
        sprintf(tmpbuf, "    Calorimetry: Target input power(w)
        fputs(tmpbuf, stdprn);
        print_now = 'n';
    } /* print due to taking calorimetry data */
    /* print data */
    /* check time for data read */
    /* check for data collection */
    /* while collecting data */
}
}
}
}

/* _setvideomode( _DEFAULTMODE ); */
} /* end of program main */

```

# REPORT DOCUMENTATION PAGE

Form Approved  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY (Leave blank)			2. REPORT DATE	3. REPORT TYPE AND DATES COVERED	
			22 DECEMBER 1992	INTERIM: 9 OCTOBER - 21 DEC., 1992	
4. TITLE AND SUBTITLE			5. FUNDING NUMBERS		
LOW VOLTAGE ARC HEAD VACUUM WELDING R & D STATUS REPORT No. 1			CONTRACT NAS8-39358 <i>7N-33-C2</i> <i>139637</i> <i>P, 33</i>		
6. AUTHOR(S)			8. PERFORMING ORGANIZATION REPORT NUMBER		
Dr. Graeme Aston					
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)			10. SPONSORING/MONITORING AGENCY REPORT NUMBER		
ELECTRIC PROPULSION LABORATORY, Inc. 1040 SYNTHES AVENUE MONUMENT, CO 80132					
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)			12. SUPPLEMENTARY NOTES		
GEORGE C. MARSHALL SPACE FLIGHT CENTER NATIONAL AERONAUTICS AND SPACE ADMINISTRATION MARSHALL SPACE FLIGHT CENTER, AL					
11. DISTRIBUTION/AVAILABILITY STATEMENT			12b. DISTRIBUTION CODE		
13. ABSTRACT (Maximum 200 words)			WORK IS DESCRIBED DETAILING THE FABRICATION OF TWO PROTOTYPE LOW VOLTAGE ARC HEADS, THE FABRICATION OF A PRECISION X,Y,Z ARC HEAD MELT TRACKING TEST FIXTURE AND SOFTWARE WRITTEN TO PERFORM AUTONOMOUS DATA COLLECTION DURING ARC HEAD TESTING.		
14. SUBJECT TERMS			15. NUMBER OF PAGES		
ARC HEAD, LOW VOLTAGE, VACUUM WELDING			32		
			16. PRICE CODE		
17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT		
EPL PROPRIETARY	UNCLASSIFIED	UNCLASSIFIED	UNLIMITED		